NOISE-ROBUST EXEMPLAR MATCHING FOR RESCORING QUERY-BY-EXAMPLE SEARCH

Emre Yılmaz[1,2]*, Julien van Hout*[2] *and Horacio Franco*[2]

[1] CLS/CLST, Radboud University, Nijmegen, Netherlands
[2] Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA

ABSTRACT

This paper describes a two-step approach for keyword spotting task in which a query-by-example (QbE) search is followed by noise robust exemplar matching (N-REM) rescoring. In the first stage, subsequence dynamic time warping is performed to detect keywords in search utterances. In the second stage, these target frame sequences are rescored using the reconstruction errors provided by the linear combination of the available exemplars extracted from the training data. Due to data sparsity, we align the target frame sequence and the exemplars to a common frame length and the exemplar weights are obtained by solving a convex optimization problem with non-negative sparse coding. We run keyword spotting experiments on the Air Traffic Control (ATC) database and evaluate performance of multiple distance metrics for calculating the weights and reconstruction errors using convolutional neural network (CNN) bottleneck features. The results demonstrate that the proposed two-step keyword spotting approach provides better keyword detection compared to a baseline with only QbE search.

Index Terms— Noise robust exemplar matching, query-by-example, keyword spotting, sparse representations

1. INTRODUCTION

Data-driven approaches to keyword spotting such as query-by-example (QbE) search have gained interest in recent years due to their ability to perform well without necessarily relying on an automatic speech recognition (ASR) system [1–4]. In these systems, a keyword is enrolled using one or several audio examples and several template-matching techniques are used to detect similar patterns in the search utterances. Not requiring knowledge of the languages of interest or language-matched training data, some QbE systems can even function in a fully language-agnostic way [5]. This research has shown that techniques leveraging supervised, discriminatively trained tokenizers, such as bottleneck features along with dynamic time warping (DTW), are among the highest-performing single systems in language-agnostic QbE under channel- and noise-degraded conditions.

Current bottleneck architectures that have been applied to QbE include a simple five-layer bottleneck that can be trained in a multilingual setting [6]. While deep neural network (DNN) models have been used in conjunction with noise-robust features to cope with channel and noise mismatch, more recently convolutional neural networks (CNN) have been introduced that use frequency convolution and pooling layers inspired from image recognition. CNNs have been shown to largely outperform standard DNNs in clean and noisy speech recognition tasks [7]. Promising keyword detection results have been reported with bottleneck features extracted using a CNN architecture in [8].

Using exemplars[1] in a sparse representation (SR) formulation provides significantly improved noise robustness and exemplar-based sparse representations have been successfully used for feature extraction, speech enhancement and noise robust speech recognition tasks [9–12]. A variant of the SR-based techniques, dubbed *noise robust exemplar matching* (N-REM), approximates the spectral representations of noisy speech segments as a superposition of speech and noise exemplars and performs reconstruction error (RE)-based decoding by applying dynamic programming [13, 14]. N-REM uses exemplars of multiple length, each corresponding to a speech unit, which are organized in separate dictionaries based on length and class (of the associated speech unit). The recognizer adopts a reconstruction error-based back-end, i.e., the recognition is performed by comparing the quality of the match for different classes quantified by a distance/divergence measure and choosing the class sequence that minimizes the total reconstruction error (RE).

In this work, we adjust the N-REM formulation for rescoring the segments which are detected during the QbE search by using the same exemplars in a sparse representation formulation. The rationale behind using sparse representations in a combined setting with the QbE search is similar to [15]. This approach differs as we use actual exemplars that are (warped) feature sequences from training data and organized in query-specific dictionaries rather than learning basis vectors in a single dictionary for modeling all queries. Class-specific dictionaries, i.e., dictionaries containing exemplars of a single speech unit, are known to provide a more precise representation of the corresponding class in the high dimensional feature space yielding better ASR performance [14]. Holding the non-negativity constraint, speech is represented in the bottleneck feature space and linear combination weights are learned to approximate target search segments in the rescoring step. Learning the weights by solving the convex optimization problem with a cost function minimizing the approximation error is expected to provide better modeling in the high-dimensional feature space compared to the exemplar averaging (with equal weights) done during the QbE search for improved keyword detection.

All exemplars belonging to a certain query are organized in a separate dictionary similar to the N-REM ASR framework. Moreover, we add artificial noise exemplars [16] to these speech dictionaries for modeling possible mismatches between the exemplars and search segments. One crucial difference in the keyword spotting scenario is the limited amount of training data, which results in very few exemplars for each query. One solution to this problem is aligning these exemplars with different length to a common length using subsequence dynamic time warping (DTW) [17] and creating a single

[1] 'Exemplars' and 'examples' are defined as (warped and/or averaged) feature sequences representing a single query in this paper. Both words are used interchangably throughout the paper.

dictionary for each query. Using these query-specific dictionaries, we assign a RE-based detection score to the search segments detected during the QbE search. We further investigate the influence of the dissimilarity measure used for calculating the RE-based detection score on the keyword detection accuracy.

This paper is organized as follows. Section 2 summarizes the baseline QbE system and the proposed rescoring scheme is presented in Section 3. The experimental setup is described in Section 4 and the recognition results are presented in Section 5. Section 6 concludes the paper.

2. QBE SYSTEM

The baseline query-by-example (QbE) system is similar in architecture to the dynamic time warping (DTW) single systems described in [8] and follows a simple architecture consisting of three steps: (1) bottleneck feature extraction, (2) DTW matching and (3) score normalization. Speech activity detection (SAD) is not used in these experiments because it was not shown to help on this data.

Due to their superior performance in QbE search, bottleneck features are used which are extracted using a CNN model trained in a multilingual fashion using 5 languages. Linguistic experts created a universal phone set to map phones from multiple languages to a unified set. Acoustic clustering of triphones is then used to create more than 5000 senones, which are used as targets for the output layer of a bottleneck network.

When several examples are available to enroll each query, two main approaches have been used to combine those examples: 1) subsequence dynamic time warping (sDTW) [17] is applied with each example separately and the detections are merged in a late-stage fashion using principled fusion or a voting system [1], and 2) the examples are merged together prior to DTW search into a single example, and sDTW is applied only using this merged example [18].

Because we are primarily concerned with speed, our baseline QbE search uses the second approach and is implemented as follows: assuming that there are N examples for a particular query in no particular order, two examples are picked randomly and aligned using standard DTW. Then, the sigmoid bottleneck features are averaged for each frame along the alignment path yielding a merged example with the same length of the longer example. The third example is aligned with the merged example in a similar way. This process is repeated until all examples available for a query recorded in the same condition are merged. For each query, the length of the final merged example is equal to the length of the longest available example.

The query search is achieved by using sDTW, with the memory-efficient improvements proposed in [19]. When using a single example to enroll, the dynamic programming algorithm is applied as follows: we initialize distances to 0 at each frame of the search utterance to enable the best paths to start anywhere. Then, we progressively compute the minimum accumulated distance through the joint distance matrix between the query and search utterance. Local path constraints only allow moving horizontally, vertically, or diagonally by one frame at a time. Path normalization by total path length is applied when making best-path decisions as well as at the end. At each step, the memory-efficient implementation only stores three values: the starting frame, the current path length, and accumulated distance. At the end of the search utterance, we look for local minima in the normalized accumulated distance of the paths going through the entire query ending at each frame. For each local minimum, we retrieve the stored starting frame for this path. Pairwise comparison of all detections for a particular query enable merging the detections overlapping by more than 50% by keeping the detection of least normalized distance. For the QbE search, we found that cosine distance performed best for CNN bottleneck features similar to [8]. In this work, we only report QbE search results using cosine distance.

Because the normalized distance $\mathrm{D}_{\mathrm{norm}}$ of any path lies in the range of [0,1], we can map the normalized distance to a detection score DS as

$$\mathrm{DS} = (1 - \mathrm{D}_{\mathrm{norm}}). \tag{1}$$

The distribution plots of these scores for each query are unimodal with variations in the means and variances depending on the query. M-norm is applied to the detection scores (the m-normalized scores are henceforth referred to as mDS) to recalibrate detections from different queries.

3. N-REM RESCORING

3.1. Modeling speech using exemplars

To improve upon the above example-merging QbE baseline, we run a first pass search using the final merged examples and rescore the most promising detections by linearly combining each example in a sparse representation formulation. Unlike the iterative averaging applied in the first pass, we can assign more reliable detection scores by learning unique exemplar weights for each test segment. Due to data sparsity, all exemplars belonging to a query are aligned to the longest available exemplar of that query. The aligned speech exemplars, each comprised of D features for each frame and spanning l frames, are reshaped into a single vector and stored in the columns of a speech dictionary $\mathbf{S}_{c,l}$: one for each query c at the length of the longest available exemplar l. Each dictionary is of dimensionality $Dl \times N_{c,l}$ where $N_{c,l}$ is the number of available speech exemplars of query c. Similarly, an artificial noise dictionary $\mathbf{N}_l$ for the same length l is formed. As described in [20], each artificial noise exemplar has non-zero entries at a single feature dimension for the complete duration of that exemplar. This leads to an artificial noise dictionary with D exemplars. These exemplars are used to cope with possible mismatches in the high-dimensional feature space for each feature dimension. Each speech dictionary is concatenated with the artificial noise dictionary of the same length to form a combined dictionary $\mathbf{A}_{c,l} = [\mathbf{S}_{c,l}\, \mathbf{N}_l]$ of dimensionality $Dl \times M_{c,l}$ where $M_{c,l}$ is the total number of available speech and noise exemplars.

Each detected search segment of length T frames is aligned to the length of the longest available exemplar l with the same label and also reshaped into a vector $\mathbf{y}_l$. For each detection, the corresponding $\mathbf{y}_l$ is expressed as a linear combination of the exemplars that are stored in the dictionary with the same label:

$$\mathbf{y}_l \approx \sum_{m=1}^{M_{c,l}} x_{c,l}^m \mathbf{a}_{c,l}^m = \mathbf{A}_{c,l}\mathbf{x}_{c,l} \qquad \text{s.t.} \qquad x_{c,l}^m \geq 0 \tag{2}$$

where $\mathbf{x}_{c,l}$ is an $M_{c,l}$-dimensional non-negative weight vector. The sparse solutions of $\mathbf{x}_{c,l}$ yield more realistic approximations of the detected search segments without overfitting and have been shown to provide better recognition results [21, 22].

The non-negative exemplar weights $\mathbf{x}_{c,l}$ are obtained by minimizing the cost function,

$$d(\mathbf{y}_l, \mathbf{A}_{c,l}\mathbf{x}_{c,l}) + \sum_{m=1}^{M_{c,l}} x_{c,l}^m \Lambda_m \qquad \text{s.t.} \qquad x_{c,l}^m \geq 0 \tag{3}$$

where $\mathbf{\Lambda}$ is an $M_{c,l}$-dimensional vector. The first term is the divergence between the detected search segment and its approximation. The second term is a regularization term that penalizes the l_1-norm of the weight vector to produce a sparse solution. $\mathbf{\Lambda}$ contains non-negative values and controls how sparse the resulting vector $\mathbf{x}$ is. By defining $\mathbf{\Lambda}$ as a vector, the amount of sparsity enforced on different types of exemplars can be adjusted. In this case, using a high sparsity factor for artificial noise exemplars is required to avoid them getting relatively high weights compared to the speech exemplars.

In this work, non-negative sparse coding (NSC) is applied to obtain the exemplar weights that minimize the cost function. We adopt the generalized Kullback-Leibler divergence (KLD) for d as both generalized KLD and Euclidean distance provide very similar approximations using the bottleneck features in pilot experiments. For the NSC solution of the exemplar weights, we apply the multiplicative update rules given by

$$\mathbf{x}_{c,l} \leftarrow (\mathbf{x}_{c,l} \odot ((\mathbf{A}_{c,l}^T(\mathbf{y}_l \oslash (\mathbf{A}_{c,l}\mathbf{x}_{c,l}))) \oslash (\mathbf{A}_{c,l}^T\mathbf{1} + \mathbf{\Lambda})) \quad (4)$$

with $\odot$, $\oslash$ and $\cdot^{[\,]}$ denoting element-wise multiplication, element-wise division and element-wise exponentiation respectively. By iteratively applying this update rule, the weight vector becomes sparse, and the reconstruction error between the vectorized detected search segment and its approximation decreases monotonically.

3.2. Recombining exemplars using sparse weights

After a fixed number of iterations, we reconstruct the approximation using the learned weights, and the RE normalized with the frame length is calculated with the generalized KLD and frame-level cosine distance. The latter is done due to the superior performance of cosine distance on CNN bottleneck features for QbE search. We convert the normalized RE (RE_{norm}) into an reconstruction error-based detection score as

$$\text{RS} = 1 - \text{RE}_{\text{norm}} + K \quad (5)$$

where K is a constant chosen to shift the RE_{norm} to a range similar to the DS values obtained in the first step. We then apply m-normalization to the RS values (mRS), using the same query-specific m-norm parameters obtained in the first step. The final detection score FS is obtained as a weighted sum of mDS and mRS for each detection,

$$\text{FS} = \text{mRS} * \text{RW} + \text{mDS} * (1 - \text{RW}) \quad (6)$$

where RW is the rescoring weight which lies in the range [0,1].

4. EXPERIMENTAL SETUP

4.1. Databases

For feature extraction, we use a CNN bottleneck network trained multilingually using speech material originating from five languages and various datasets: (1) Dari (TransTac); (2) Egyptian Arabic (CALLHOME); (3) English (Fisher); (4) Mandarin (GALE); and (5) Spanish (CALLHOME). All data is sampled at 8 kHz. More details regarding the training material are found in [23].

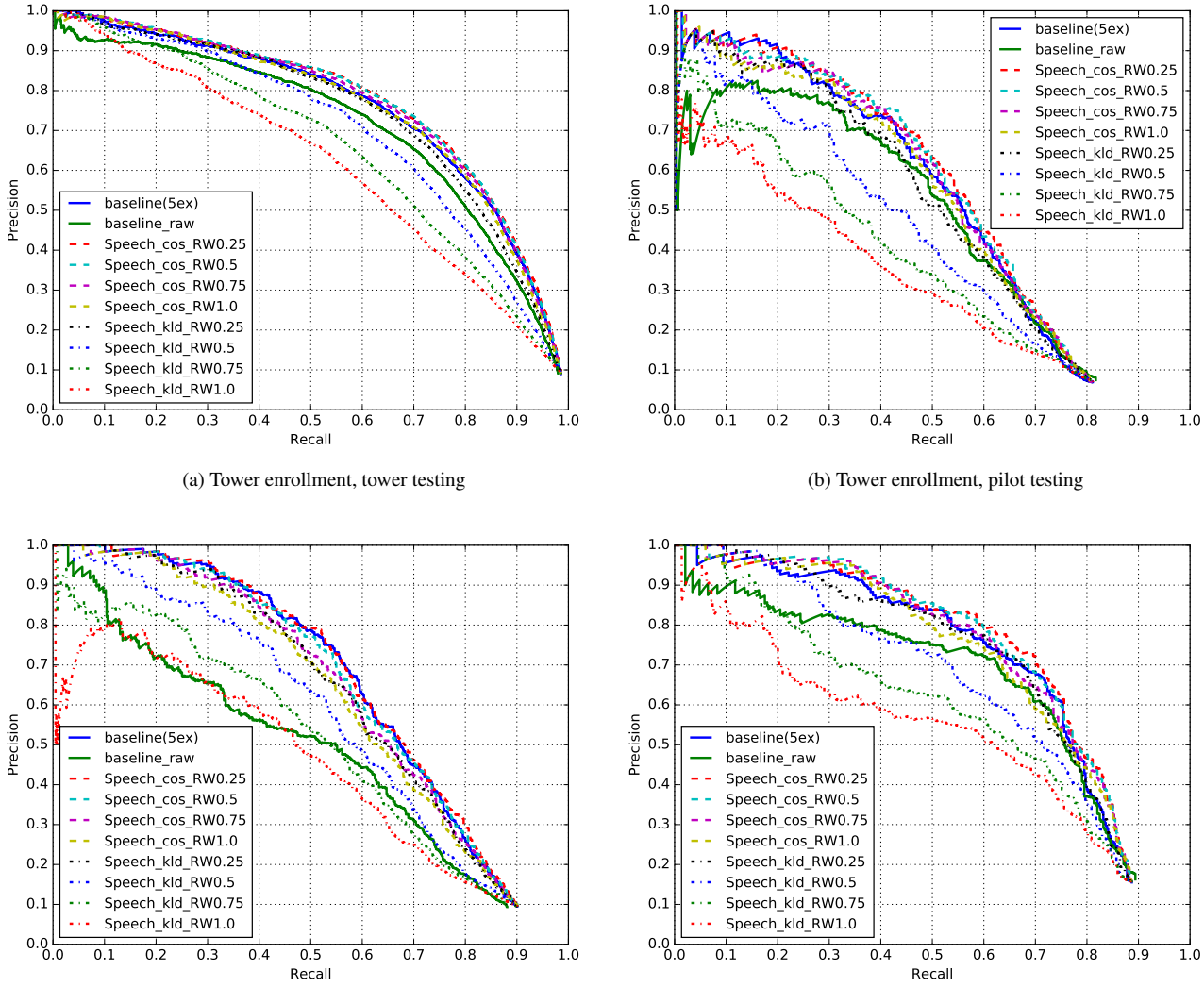
We use the Air Traffic Control (ATC) database [24] to investigate the performance of the proposed rescoring technique for keyword spotting task. This database includes voice communication traffic between various controllers and pilots at three US airports, and was initially released for ASR under LDC catalog number LDC94S14A. We subsequently split the data into parts used for query enrollment and for search. A total of 167 frequent queries such as "join the localizer" or "maintain three thousand" were selected, and forced alignment based on the ASR reference are used to obtain word-level alignments for enrollment as well as to score the search. Each query had an average of 20 examples available for enrollment.

The data was split into two sets: the audio originating from the control towers (32 distinct speakers) and the audio originating from the pilots (283 distinct speakers). Both sets have a large degree of corruption and acoustic variations but the latter is particularly corrupted and of particularly low degree of intelligibility due to the distance between the plane and the receiver, engine noise, weather, breathing from the pilot into the microphone, etc. In this work, we run keyword spotting experiments in four evaluation conditions: (1) enrolling and testing with speech from control towers (tower enrollments, tower testing); (2) enrolling with speech from control towers and testing with speech from pilots (tower enrollments, pilot testing); (3) enrolling with speech from pilots and testing with speech from control towers (pilot enrollments, tower testing) and (4) enrolling and testing with speech from pilots (pilot enrollments, pilot testing).

4.2. Implementation details

The CNN trained for feature extraction consists of five hidden layers, with 1024 neurons in each layer and a bottleneck in its 3$^{\text{rd}}$ hidden layer that consists of 60 neurons. It uses 200 filters of dimension 8, with a max-pooling of 3. All hidden layers use sigmoid activations. The bottleneck system is trained using time-domain gammatone filterbank (TDGFB) features. The TDGFB features are extracted by using SRI's gammatone filterbank implementation, in which a bank of time-domain gammatone filters consisting of 40 channels is used, with the filters equally spaced on the equivalent rectangular bandwidth (ERB) scale. The TDGFBs consist of filterbank energies computed over an analysis window of 25.6 ms, at a frame advance of 10 ms, with the energies root-compressed using the 15th power root followed by an utterance-level mean subtraction across each feature dimension. To tackle noise and channel mismatch, we apply cepstral mean subtraction (CMS) on the utterance level and mean-variance normalization (MVN) on the corpus level to the TDGFB features. A sigmoid transformation is applied to the raw activations which reduces the dynamic range and gives better results than using the raw activations with the cosine distance. We found that using SAD to drop non-speech frames for enrollment did not help improve performance.

For rescoring, we created speech dictionaries for each query in a given recording condition containing a fixed number of the exemplars that are used in the first step. In our experiments, we rescore the detection results that have been obtained by using 5 and 10 merged examples in the QbE search. The same exemplars are aligned to the length of the longest available exemplar and stacked in a dictionary for rescoring. We use 197 speech dictionaries with 5 exemplars and 138 speech dictionaries with 10 exemplars, each containing exemplars representing different queries recorded in a certain condition. The columns of the speech and noise dictionaries and vectorized search vectors are l_2-normalized. The speech exemplars are represented as bottleneck features with $D = 60$. This results in 60 artificial noise exemplars each having a fixed value of 1 for each dimension. The exemplar weights are obtained after 100 iterations of the multiplicative update rule. Elements of $\mathbf{\Lambda}$ in Equation (4) are set to 1.5 and 4 for speech and noise exemplars respectively. The constant K in Equation (5) is set to 0.1. As the sDTW algorithm outputs several detections per utterance for each query, we can only rescore



(a) Tower enrollment, tower testing

(b) Tower enrollment, pilot testing

(c) Pilot enrollment, tower testing

(d) Pilot enrollment, pilot testing

Fig. 1: Precision-recall curves obtained using speech dictionaries only - using 5 speech exemplars

a small subset of these detections in order to keep the computation tractable. We apply a threshold of 3 on the m-norm scores, which effectively only selects the scores that are 3 standard deviations above the median score for each query. In total, we rescore 119 793 detections obtained using 5 exemplars and 76 019 detections obtained using 10 exemplars during the QbE search.

4.3. Keyword spotting experiments

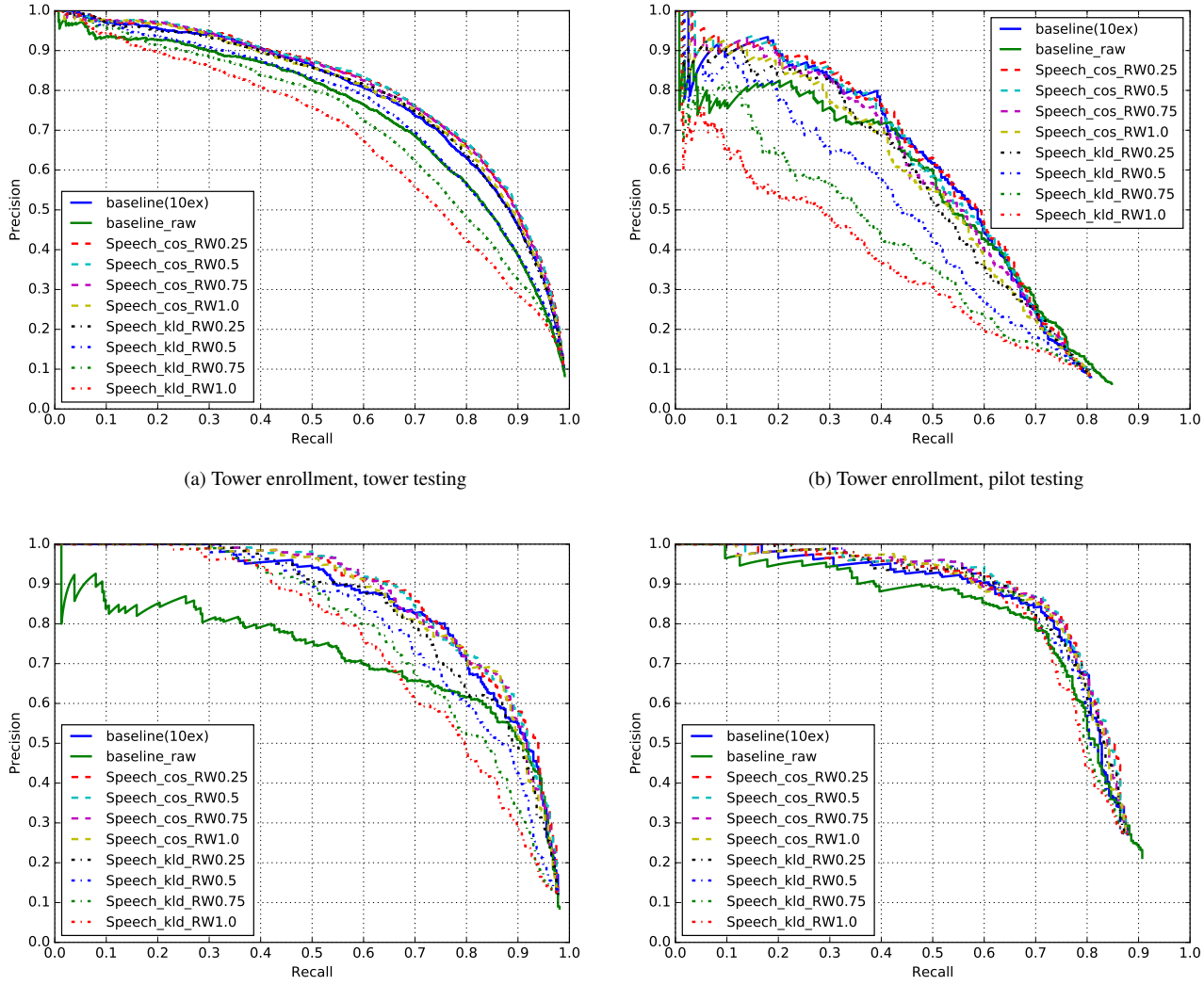
We run the baseline QbE system to obtain the first pass detection scores using merged examples and perform rescoring using the exemplar matching system as described in Section 3. We plot precision-recall curves for visualizing the performance of each system. The performance of the raw (baseline_raw) and m-normalized (baseline(5ex) or baseline(10ex)) QbE scores are presented as the baseline curves. We perform rescoring on the detection scores obtained using 5 and 10 merged examples to be able to explore the impact of the amount of available exemplars on the keyword spotting performance.

In the first set of experiments, we rescore detections with the N-REM system using 5 speech exemplars only and we evaluate the impact of the dissimilarity measure and the rescoring weight. The RE is calculated with the generalized KLD (kld) and cosine distance (cos) to compare the keyword detection performance of these metrics. Furthermore, we vary the rescoring weight from 0 (only mDS considered) to 1 (only mRS considered) with steps of 0.25. In the second set of experiments, we increase the number of exemplars to 10 and report the detection results with the same metrics and RWs used in the initial experiments. Finally, we use dictionaries including both speech and artificial noise dictionaries. The same detections are rescored and compared with the results obtained in the first set of experiments to see if rescoring benefits from using artificial noise exemplars.

5. RESULTS AND DISCUSSION

5.1. Different Dissimilarity Measures and Rescoring Weights

The detection results for the first set of experiments using 5 speech exemplars are presented in Figure 1. The curve for the baseline QbE



(a) Tower enrollment, tower testing

(b) Tower enrollment, pilot testing

(c) Pilot enrollment, tower testing

(d) Pilot enrollment, pilot testing

Fig. 2: Precision-recall curves obtained using speech dictionaries only - using 10 speech exemplars

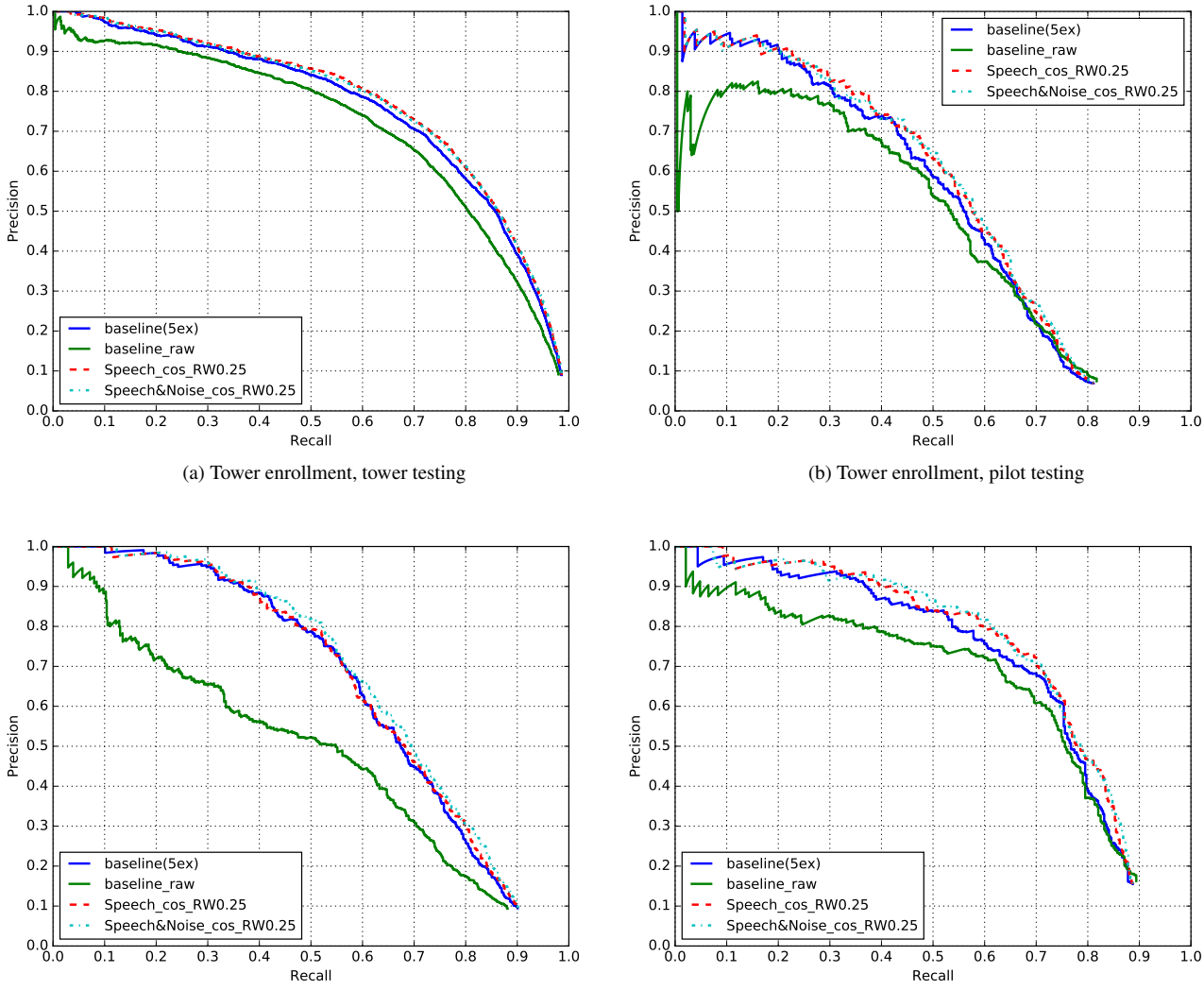
system (RW = 0) is marked in blue block line. In all conditions and for all RWs, the systems using mRS calculated with the cosine distance (dashed curves) perform better than their counterparts using the generalized KLD (dashed dotted curves). Moreover, rescoring using the cosine distance brings varying amounts of improvements compared to the baseline QbE system in all conditions. Even using only speech exemplars in the dictionaries, the rescoring yields moderate improvements in enrollments and testing with pilot speech as shown in Figure 1b, 1c and 1d. Although the impact of RW on the keyword detection performance becomes less visible in some conditions, combining the QbE and N-REM scores with a RW = 0.25 (red dashed curve) seems to provide the highest improvement in the keyword detection performance in general.

5.2. Using More Speech Exemplars

The detection results obtained by using 10 speech exemplars are presented in Figure 2. Doubling the number of speech exemplars, the rescoring provide better detection results with pilot enrollments as shown in Figure 2c and 2d compared to Figure 1c and 1d. This is expected to be a consequence of the increased number of speech exemplars helping to cope with the increased variation in enrollments with pilot speech and yielding more confident detection scores.

5.3. Using Artificial Noise Exemplars

In the last set of experiments, we add artificial noise exemplars to the dictionaries used for rescoring. Due to space limitations and similar results, we only present the results obtained by using 5 speech exemplars. The keyword spotting system using the cosine distance and combining mDS and mRS values with a RW = 0.25 as the final detection score outperforms the rest of the systems in the initial experiments. Therefore, we only report the results obtained using this system for the clarity of the plots. The comparison of the systems using speech dictionaries only (dashed curve) and speech and noise dictionaries (dashed dotted curve) for approximating search segments is illustrated in Figure 3. From Figure 3a, it is evident that using artificial noise exemplars does not help with the detection



(a) Tower enrollment, tower testing

(b) Tower enrollment, pilot testing

(c) Pilot enrollment, tower testing

(d) Pilot enrollment, pilot testing

Fig. 3: Precision-recall curves obtained using speech and artificial noise dictionaries - using 5 speech exemplars

accuracy in tower enrollment and testing condition. On the other hand, for pilot enrollments, the systems using noise exemplars appear to perform slightly better detection as shown in Figure 3c and 3d. From these results, we can conclude that adding artificial noise exemplars slightly increases the robustness of the approximation in enrollments with pilot speech.

6. CONCLUSIONS

We describe a noise-robust exemplar matching-based rescoring scheme for query-by-example search for keyword spotting. The baseline QbE systems use merged examples obtained by aligning and averaging the exemplars belonging to the same query recorded under similar conditions. In our approach, we apply subsequence DTW, and a list of possible detections is created in the first step. The proposed rescoring system uses the same exemplars to linearly approximate these detections and to output a reconstruction error-based detection score. By combining the detection scores provided in the first and rescoring step with a rescoring weight, the detection performance of a QbE system can be improved considerably for tower enrollments and mildly for pilot enrollments. We further observed that using cosine distance for calculating the reconstruction error-based detection scores provides better detection accuracy compared to using generalized KLD. Finally, using artificial noise exemplars mildly improves the keyword spotting performance using pilot enrollments.

7. ACKNOWLEDGEMENTS

This work was performed while the first author was on a research visit to SRI International. This visit has been funded by the NWO Project 314-99-119 (Frisian Audio Mining Enterprise) and SRI International. The second author's work is partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0037. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. "A" (Approved for Public Release, Distribution Unlimited).

8. REFERENCES

[1] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, 2009, pp. 421–426.

[2] Zhang Y. and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, 2009, pp. 398–403.

[3] J. Proença, A. Veiga, and F. Perdigão, "Query by example search with segmented dynamic time warping for non-exact spoken queries," in *Proc. EUSIPCO*, 2013, pp. 1691–1695.

[4] X. Anguera, L. J. Rodríguez-Fuentes, A. Buzo, F. Metze, I. Szöke, and M. Peñagarikano, "QUESST2014: Evaluating query-by-example speech search in a zero-resource setting with real-life queries," in *Proc. ICASSP*, 2015, pp. 5833–5837.

[5] H. Xu, P. Yang, X. Xiao, L. Xie, C. C. Leung, Hongjie Chen, Jia Yu, Hang Lv, L. Wang, S. J. Leow, B. Ma, E. S. Chng, and H. Li, "Language independent query-by-example spoken term detection using n-best phone sequences and partial matching," in *Proc. ICASSP*, April 2015, pp. 5191–5195.

[6] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2012, pp. 336–341.

[7] Ossama Abdel-Hamid, Abdel-Rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[8] I. Szoke, M. Skacel, L. Burget, and J. H. Cernocky, "BUT QUESST 2014 System Description," in *Proc. MediaEval 2014 Workshop*, 2014.

[9] T. N. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, and A. Sethy, "Sparse representations features for speech recognition," in *Proc. INTERSPEECH*, Sept. 2010, pp. 2254–2257.

[10] J.F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2067–2080, Sept. 2011.

[11] A. Hurmalainen, J.F. Gemmeke, and T. Virtanen, "Non-negative matrix deconvolution in noise robust speech recognition," in *Proc. ICASSP*, May 2011, pp. 4588–4591.

[12] Q. F. Tan and S. S. Narayanan, "Novel variations of group sparse regularization techniques with applications to noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1337–1346, May 2012.

[13] E. Yılmaz, J. F. Gemmeke, D. Van Compernolle, and H. Van hamme, "Noise-robust digit recognition with exemplar-based sparse representations of variable length," in *IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, Santander, Spain, Sept. 2012, pp. 1–4.

[14] E. Yılmaz, J. F. Gemmeke, and H. Van hamme, "Noise robust exemplar matching using sparse representations of speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22(8), pp. 1306–1319, Aug. 2014.

[15] D. Ram, A. Asaei, and H. Bourlard, "Subspace detection of DNN posterior probabilities via sparse representation for query by example spoken term detection," in *Proc. INTERSPEECH*, 2016, pp. 918–922.

[16] J. F. Gemmeke and T. Virtanen, "Artificial and online acquired noise dictionaries for noise robust ASR," in *Proc. INTERSPEECH*, 2010, pp. 2082–2085.

[17] M. Müller, *Information Retrieval for Music and Motion*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[18] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "High-performance query-by-example spoken term detection on the SWS 2013 evaluation," in *Proc. ICASSP*, May 2014, pp. 7819–7823.

[19] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for Query-by-Example spoken term detection," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, July 2013, pp. 1–6.

[20] J.F. Gemmeke and T. Virtanen, "Noise robust exemplar-based connected digit recognition," in *Proc. ICASSP*, March 2010, pp. 4546–4549.

[21] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.

[22] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, March 2007.

[23] C. Bartels, W. Wang, V. Mitra, C. Richey, A. Kathol, D. Vergyri, H. Bratt, and C. Hung, "Toward human-assisted lexical unit discovery without text resources," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2016, pp. 64–70.

[24] J. Godfrey, "Air Traffic Control Complete LDC94S14A," 1994, Web Download. Philadelphia: Linguistic Data Consortium.